IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): Vassey	
Application No.: 10/598,654	Group Art Unit: 2178
Filed: 03/21/2007	Examiner: Stork
Title: Script Generation	
Attorney Docket No.: 760-022	

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANT'S REPLY BRIEF

Sir:

Please enter this Reply Brief.

I. **Real Party in Interest**

The real party in interest is Business Integrity Limited.

II. **Related Appeals and Interferences**

Appellants are not aware of any related appeals or interferences.

III. **Status of the Claims**

Claims 1-29 are currently pending in this application. All of the pending claims are rejected. No claims have been allowed. The rejections of independent claims 1, 15 and 16 are the subject of this appeal.

IV. **Status of Amendments**

All submitted amendments have been entered and considered.

V. **Summary of Claimed Subject Matter**

The presently claimed invention is related to the sequence in which update functions are invoked by automated document generation systems. Automated document generation systems typically generate a customized document such as a legal contract by using a template and answers submitted in response to a questionnaire form. For example, a user without legal expertise would input answers to questions posed by the questionnaire, and those answers would be used to generate a customized contract document from the template by selecting particular clauses of legal significance to include or exclude along with other content that may always be included. One of the advantages of such systems is reduced reliance on skilled personnel such as lawyers. However, complex relationships between questions and answers can cause problems. It is now known that different questions can be related in the sense that the answer to one question may determine or change the possible answers to another question, or invalidate the other question entirely. For example, if the answer to the question "gender?" is "male" then the answer to "title?" must be "Mr." and the question "pregnant?" is invalid. Similarly, if the answer to "title?" is "Mr." then "gender?" is "male" and "pregnant?" is invalid. However, *answers may also be related* and *the order in which the answers to questions are used can affect the result.* For example, if "state of incorporation?" is Delaware and "state of residence?" is Massachusetts then a jurisdiction clause selected based on the question "jurisdiction?" may end up being **either** Delaware **or** Massachusetts based on the order in which the questions are answered.[1] The presently claimed invention mitigates such unacceptably random results by determining the sequence in which to invoke update functions by identifying the trigger questions and then using an activation network to select the invocation sequence for the update functions associated with those trigger questions.

The limitations recited in the independent claims are supported by the specification and drawing as indicated in bold below.

[1] More typically, each question is assigned a value based on its answer, and the order in which those values are applied to the template (invoked) after all of the questions have been asked and answered causes the difficulty rather than the order in which the questions are asked or answered. The order of answering is used to illustrate the problem because it conveys the same principle in more easily understandable terms, i.e., what would happen if update functions were invoked as questions were answered.

1. (previously presented) A method of generating an invocation sequence of update functions that update values associated with questions in a questionnaire form used to obtain information to produce a customized document from a template document, the invocation sequence specifying an order in which the update functions are invoked,

The form in Figure 1a illustrates a typical form that may be generated by a document generation program. The example chosen is that of providing details of a buyer in a purchase contract. Similar forms may be generated by other types of form generators, such as database engines, for example, Oracle 8i™. These forms all have an element of dynamic behaviour in common, in that answering a question or providing a variable or value affects other questions appearing on the form. Page 8: 23 through page 9: 3.

the method comprising the steps of:

supplying the update functions to a synthesiser;

The synthesis of OnChange functions can occur at one of several points in the process of rendering a form for final use by a user. For clarity, the code generation engine which carries out the synthesis of the OnChange functions will be referred to as an OnChange synthesiser. The function of the onChange synthesiser is to take the form description, validation functions (if any) and update functions generated by a form generator, convert these into a form description and OnChange functions, and export these in a script suitable for interpretation by a form renderer, such as a web browser. Page 21:20-28

identifying, using the synthesizer, trigger questions from the questions of the questionnaire form that trigger invocation of at least one of the supplied update functions;

The form questions are known as elements, where an element accepts input from or by a user. When the user selects the buyer status (by inputting to the appropriate element), code must be generated to perform the function of enabling or disabling the relevant address question. The function that must be carried out is known as an onChange function. There are two types of onChange function: validation function: alerts the user that the value of an element (or a combination of element values) is invalid; update function; reacts to change the value of an element by updating other elements of the form. Additionally, it is possible to define two types of element: trigger element: any element occurring within a validation function or which causes a change to another element through an update function; activated element: any element changed by an update function. Page 9: 4-17.

generating an activation network based on the identified update functions at the synthesiser; and

In order to determine these invocation sequences an activation network is constructed from the form elements (the nodes on the network) and the update functions (the arrows connecting the nodes). Page 11: 21-25.

determining the invocation sequence of the identified update functions for each trigger question by using the activation network.

The sequence in which update functions are invoked (the invocation sequence) for each trigger element is determined by a breadth-first expansion of the activation network. In this expansion, any acyclic path of length N from a first node X to a second node Y is expanded into a path of length N+1 from the first node X to a third node Z by appending a path of length 1 from the second node Y to the third node Z. Page 12: 17-22. The order in which each update is carried out is determined by the condition that all functions which lead to an update of an element are carried out before that element is subsequently used to update another: all functions corresponding to arrows entering a form element must be completed before the function corresponding to an arrow leaving the form element can be executed. Page 13: 1-5.

15. (previously presented) A computer readable medium having program code stored thereon, which, when run on a computer, causes the computer to generate invocation sequences of update functions to update values associated with questions in a questionnaire form used to obtain information to produce a customized document from a template document by performing the steps of:
The form in Figure 1a illustrates a typical form that may be generated by a document generation program. The example chosen is that of providing details of a buyer in a purchase contract. Similar forms may be generated by other types of form generators, such as database engines, for example, Oracle 8i™. These forms all have an element of dynamic behaviour in common, in

that answering a question or providing a variable or value affects other questions appearing on the form. Page 8: 23 through page 9: 3.

supplying the update functions to a synthesiser;

The synthesis of OnChange functions can occur at one of several points in the process of rendering a form for final use by a user. For clarity, the code generation engine which carries out the synthesis of the OnChange functions will be referred to as an OnChange synthesiser. The function of the onChange synthesiser is to take the form description, validation functions (if any) and update functions generated by a form generator, convert these into a form description and OnChange functions, and export these in a script suitable for interpretation by a form renderer, such as a web browser. Page 21: 20-28

identifying, with the synthesizer, trigger questions from the questions of the form that trigger the invocation of ones of the supplied update functions;

The form questions are known as elements, where an element accepts input from or by a user. When the user selects the buyer status (by inputting to the appropriate element), code must be generated to perform the function of enabling or disabling the relevant address question. The function that must be carried out is known as an onChange function. There are two types of onChange function: validation function: alerts the user that the value of an element (or a combination of element values) is invalid; update function; reacts to change the value of an element by updating other elements of the form. Additionally, it is possible to define two types of element: trigger

element: any element occurring within a validation function or which causes a change to another element through an update function; activated element: any element changed by an update function. Page 9: 4-17.

generating an activation network based on the update functions identified at the synthesiser; and

In order to determine these invocation sequences an activation network is constructed from the form elements (the nodes on the network) and the update functions (the arrows connecting the nodes). Page 11: 21-25.

determining the invocation sequence of update functions for each trigger question by using the activation network.

The sequence in which update functions are invoked (the invocation sequence) for each trigger element is determined by a breadth-first expansion of the activation network. In this expansion, any acyclic path of length N from a first node X to a second node Y is expanded into a path of length N+1 from the first node X to a third node Z by appending a path of length 1 from the second node Y to the third node Z. Page 12: 17-22. The order in which each update is carried out is determined by the condition that all functions which lead to an update of an element are carried out before that element is subsequently used to update another: all functions corresponding to arrows entering a form element must be completed before the function corresponding to an arrow leaving the form element can be executed. Page 13: 1-5.

16. (previously presented) A computer readable medium having program code stored thereon for generating an invocation sequence to update values associated with questions in a questionnaire form, comprising:

The form in Figure 1a illustrates a typical form that may be generated by a document generation program. The example chosen is that of providing details of a buyer in a purchase contract. Similar forms may be generated by other types of form generators, such as database engines, for example, Oracle 8i™. These forms all have an element of dynamic behaviour in common, in that answering a question or providing a variable or value affects other questions appearing on the form. Page 8: 23 through page 9: 3.

a synthesiser for generating an activation network based on update functions supplied by a form generator and identifying trigger questions from the questions of the form that trigger the invocation of ones of the update functions; and

The synthesis of OnChange functions can occur at one of several points in the process of rendering a form for final use by a user. For clarity, the code generation engine which carries out the synthesis of the OnChange functions will be referred to as an OnChange synthesiser. The function of the onChange synthesiser is to take the form description, validation functions (if any) and update functions generated by a form generator, convert these into a form description and OnChange functions, and export these in a script suitable for interpretation by a form renderer, such as a web browser. Page

21: 20-28. The form questions are known as elements, where an element accepts input from or by a user. When the user selects the buyer status (by inputting to the appropriate element), code must be generated to perform the function of enabling or disabling the relevant address question. The function that must be carried out is known as an onChange function. There are two types of onChange function: validation function: alerts the user that the value of an element (or a combination of element values) is invalid; update function; reacts to change the value of an element by updating other elements of the form. Additionally, it is possible to define two types of element: trigger element: any element occurring within a validation function or which causes a change to another element through an update function; activated element: any element changed by an update function. Page 9: 4-17.

a determinator for determining the invocation sequence of the supplied update functions, for each trigger question, the form being updated based on the determined invocation sequence and output in tangible format.

In order to determine these invocation sequences an activation network is constructed from the form elements (the nodes on the network) and the update functions (the arrows connecting the nodes). Page 11: 21-25. The sequence in which update functions are invoked (the invocation sequence) for each trigger element is determined by a breadth-first expansion of the activation network. In this expansion, any acyclic path of length N from a first node X to a second node Y is expanded into a path of length N+1 from the first node X to a third node Z by appending a path of length 1 from the

second node Y to the third node Z. Page 12: 17-22. The order in which each update is carried out is determined by the condition that all functions which lead to an update of an element are carried out before that element is subsequently used to update another: all functions corresponding to arrows entering a form element must be completed before the function corresponding to an arrow leaving the form element can be executed. Page 13: 1-5.

VI. Grounds of Rejection to be Reviewed on Appeal

A. Claims 1-29 are rejected under 35 U.S.C. 103(a) as being obvious based upon EP 1100013 (Maes) in view of US 6314415 (Mukherjee).

VII. Argument

A. The cited combination does not teach determining an invocation sequence of update functions for each trigger question by using an activation network where the update functions update values associated with questions in a questionnaire form used to obtain information to produce a customized document from a template document.

Three basic criteria must be met in order to establish a *prima facie* case of obviousness. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings.

Second, there must be a reasonable expectation of success. Third, the prior art references must teach or suggest all the claim limitations. (MPEP §2143).

Maes describes a "conversational markup language" which translates "conversational gestures." In other words, content originally presented in one markup language is translated into a different markup language so that it can be presented on a different device, e.g., rendering a web page on a cell phone. The translation may change the way content is presented because different devices and markup languages have different limitations, e.g., phones tend to have smaller screens than PCs. Mukherjee describes dynamic GUI generation where selection of features for display in the translated version is automated. The examiner must consider each claim as a whole, but with regard to the novel and non-obvious step of "determining the invocation sequence of the identified update functions for each trigger question by using the activation network" the rejection relies upon Maes at 0018, 0022 and 0108-0109, and Mukherjee at figures 3A-3L and column 2:19 through column 3:5.

In contrast with the cited references, the pending claims are related to the sequence in which update functions are invoked by automated document generation systems. Automated document generation systems typically generate a customized document such as a legal contract by using a template and answers submitted in response to a questionnaire form. For example, a user without legal expertise would input answers to questions posed by the questionnaire, and those answers would be used to generate a document from the template by selecting particular clauses of legal significance to include or exclude along with other content that may always be included. Questionnaires have been used for other purposes and it is now known that different questions can be related in the sense that the answer to one question may determine or change the possible answers to another question, or invalidate the other question entirely. For example, if the answer to the question "gender?" is "male" then the answer to "title?" must be "Mr." and the question "pregnant?" is invalid. Similarly, if the answer to "title?" is "Mr." then "gender?" is "male" and "pregnant?" is invalid. This basic condition is described in the specification[2] as follows:

> the form in Figure 1a illustrates a typical form that may be generated by a document generation program. The example chosen is that of providing details of a buyer in a purchase contract. Similar forms may be generated by other types of form generators, such as database engines, for example, Oracle 8i™. These forms all have an element of dynamic behaviour in common, in that answering a question or providing a variable or value affects other questions appearing on the form.

However, simply knowing that questions are related is insufficient to avoid potential problems because *answers may also be related* and *the order in which the answers to questions are used can affect the result*. For example, if "state of incorporation?" is Delaware and "state of residence?" is Massachusetts then a jurisdiction clause selected based on the question "jurisdiction?" may end up being **either** Delaware **or** Massachusetts based on the order in which the questions are answered.[3] This is described in the specification[4] as follows:

[2] Page 8, line 23 through page 9, line 3
[3] More typically, each question is assigned a value based on its answer, and the order in which those values are applied to the template (invoked) after all of the questions have been asked and answered causes the difficulty rather than the order in which the questions are asked or answered. The order of answering is

> The problem to be addressed is in what sequence all the update functions are invoked for each different trigger element. In order to determine these invocation sequences an activation network is constructed from the form elements (the nodes on the network) and the update functions (the arrows connecting the nodes).

The claimed invention mitigates such unacceptably random results by determining the sequence in which to invoke update functions by identifying the trigger questions and then using an activation network to select the invocation sequence for the update functions associated with those trigger questions. As described in the specification[5]:

> The order in which each update is carried out is determined by the condition that all functions which lead to an update of an element are carried out before that element is subsequently used to update another: all functions corresponding to arrows entering a form element must be completed before the function corresponding to an arrow leaving the form element can be executed.

One embodiment of this feature is described in the specification[6] as:

> The sequence in which update functions are invoked (the invocation sequence) for each trigger element is determined by a breadth-first expansion of the activation network. In this expansion, any acyclic path of length N from a first node X to a second node Y is expanded into a path of length N+1 from the first node X to a third node Z by

used to illustrate the problem because it conveys the same principle in more easily understandable terms, i.e., what would happen if update functions were invoked as questions were answered.

[4] Page 11, lines 21-25

[5] Page 13, lines 1 through 5

[6] Page 12, lines 17 through 22

appending a path of length 1 from the second node Y to the third node Z.

It will therefore be appreciated that the invention recited in the pending claims not only determines whether questions are related but also determines *an invocation sequence for applying the answers to those questions*. Consequently, creating a questionnaire or adding questions to an existing questionnaire is enhanced via automation.

Turning now to the rejections, there is no suggestion in the references of determining an invocation sequence of update functions for trigger questions by using an activation network. Paragraph 0018 of Maes simply describes how different modalities (browsers) are synchronized, e.g., when a user provides input via mobile browser "A," that input is immediately translated into corresponding input for standard browser "B." Note that the order in which that input is supplied is driven manually by the user, and the order in which input is invoked is not even considered. Even the previous characterization of the examiner that "upon identifying an update function, the CML interpreter determines the CML function to be invoked to handle the update" does not meet the recited limitation of determining an *invocation sequence* because finding *the* CML function corresponding to an update function does not yield a *sequence of functions*. Mukherjee recognizes that questions can be related but fails to recognize that the order in which the answers to questions are used can also change the result. Mukherjee also fails to describe determining how answers to questions are hierarchically related. In sum, neither reference provides any suggestion of determining an invocation sequence of identified update functions for each trigger question by using an activation network.

In the **Response To Arguments** the examiner counters that Maes discloses identification of trigger elements from the elements of the form in the update functions that trigger the invocation of the update function at paragraphs 0018, 0022 and 0061, i.e., the selection "order drink" causes invocation of the update function. Paragraph 0018 is already discussed above. Paragraph 0022 similarly describes cosmetic altering of the presentation so it can be ported to a different device. Paragraph 0061 and figure 3 describe a specific example where one question on a web page is converted to either voice for presentation on a desktop phone or to a format suitable for display on a cell phone screen. The paragraph and figure cannot possibly support the rejection because only one question is presented. There can be no sequence of questions determined when there is only one question. Recall that the presently claimed invention is directed to problems caused by answers to *different questions* being hierarchically related or the order in which answers to *different questions* are used affecting the result. In the context of Maes for example, and ignoring the context of document generation recited in the pending claims, if a first question prompted the drink selection and a second question prompted a meal selection and the selected meal included a drink other than that selected in response to the first question then there could be some ambiguity about which drink the person actually wanted. Such ambiguity is not created when one simple question is presented as described by Maes in paragraph 0061 and figure 3.

In addition to the distinctions described above it should be appreciated by the BPAI that the cited combination of references fails to result in an automated document generation system. The combination of references lacks a questionnaire with questions that prompt answers that are used to select content to produce a customized document from a template document. The preamble of each of the independent claims in view of the respective claim body describes the context of the claimed invention as a document generation system where update values associated with questions in a questionnaire form are used to obtain information to produce the customized document from a template document. The cited references cannot even be reasonably combined or modified to result in a general system for producing a customized document from a template document and questionnaire form as recited in the preamble of claim 1. The cited references describe different technology for solving different problems to achieve different goals. Note for example that Maes *modifies the presentation* of a question rather than using the answer to a question to *select content* from a template to produce a customized document. During examination, statements in the preamble reciting the purpose or intended use of the claimed invention must be evaluated to determine whether the recited purpose or intended use results in a structural difference (or, in the case of process claims, manipulative difference) between the claimed invention and the prior art. If so, the recitation serves to limit the claim. See, e.g., *In re Otto*, 312 F.2d 937, 938, 136 USPQ 458, 459 (CCPA 1963). As indicated by the claim language, the pending claims are related to the sequence in which update functions are invoked by automated document generation systems. This further distinguishes the pending claims from the cited combination.

In the **Response To Arguments** the examiner also counters that use of an acyclic graph and breadth-first expansion to determine the order in which update functions are invoked is not disclosed in the claims. However, applicant did not argue that such limitations were in the claims. Applicant quoted passages from the specification so that the examiner might appreciate the context of the invention and what is involved in at least one embodiment of determining the invocation sequence of the identified update functions for each trigger question by using the activation network. With a more complete appreciation of the claimed invention it would be apparent that the cited references describe different technology for solving different problems to achieve different goals and do not teach the limitations recited in the claims.

The BPAI should note that the claims also distinguish the cited combination by reciting use of activation networks for determining the invocation sequence of update functions. The use of activation networks offers an advantage over prior art methods of form encoding. The coding necessary to describe a form element is based on the functions necessary to create a change to or update of that form element, and the functions that must be executed once the form element is updated. Consequently, each form element needs only to be considered in relation to those form elements which either directly affect it, or are directly affected by it. This enables editing of the form in relation to one particular form element or adding or removing a form element without needing to re-encode the entire form.

In the **Examiner's Answer** the examiner counters that Maes teaches determining an invocation sequence of update functions for each trigger element by using an activation network based on the description in paragraph 0061 of a person being queried to provide age in response to ordering a drink. In other words, the examiner asserts that following a question "what would you like to drink?" with a question "what is your age?" when the answer to the first question is an alcoholic drink is equivalent to determining an invocation sequence of update functions for each trigger element by using an activation network. Appellant respectfully disagrees. There is no activation network in the cited technique, and more importantly asking a person to provide their age in response to ordering a drink is simply recognizing that two questions are *related*, which has long been known, as already described above. The inventors have recognized that simply knowing that *questions are related* is insufficient to avoid potential problems because *answers may also be related* and *the order in which the answers to questions are used can affect the result.* Using the example in Maes cited by the examiner, consider the situation where one rule is that a person must be at least 21 years old to be served alcohol, and another rule is that a member of the military can be served alcohol regardless of age. The corresponding questions might be: Are you at least 21 years old? and Are you a member of the military? Note that for an 18 year old member of the military the result (and whether or not the person will be served) differs depending on the order in which the questions are asked. For example:

Invocation Sequence 1

Are you at least 21 years old?	No	(result=do not serve)
Are you a member of the military?	Yes	(result changed to do serve)

Invocation Sequence 2

Are you a member of the military?	Yes	(result= do serve)
Are you at least 21 years old?	No	(result changed to do not serve)

Note that both of the questions are related to the question "would you like something to drink," and *also related to each other* such that invocation sequence 1 yields the correct result and invocation sequence 2 yields an incorrect result. The cited references fail to recognize the problem or suggest any solution.

Also in the **Examiner's Answer** the examiner counters that Mukherjee teaches determining an invocation sequence of update functions for each trigger element by using an activation network based on the description in figures 3A-3L and column 2, line 19 through column 3 line 5. According to the examiner the cited passage teaches that "based upon a user's answer to trigger questions, various questions are updated to become available/disabled." Appellant respectfully disagrees. There is no activation network in the cited technique, and more importantly enabling and disabling questions based upon answers is simply recognizing that two questions are *related*, which has long been known, as already described above. The inventors have recognized that simply knowing that *questions are related* is insufficient to avoid potential problems because *answers may also be related* and *the order in which the answers to questions are used can affect the result*. Consider the example in the previous paragraph. The Board will note that the problem is solved by determining the proper invocation sequence rather than enabling or disabling questions.

The examiner also asserts that "determining that questions within the sequence should be skipped based on being disabled constitutes determining an updated invocation sequence by using an activation network." Appellant respectfully disagrees. Disabling a question is not equivalent to determining placement in an invocation sequence because the disabled question is removed from the sequence (not placed). Presumably the examiner would assert that the problem in the context of Maes described above could be solved by deleting the question "Are you at least 21 years old?" if the answer to the question "Are you a member of the military?" is yes. However, the question "Are you at least 21 years old?" may be relevant to some other conditional content, e.g., how much the person is charged for their drink or where they are seated, so it cannot simply be skipped. Consequently, deleting the question causes another problem whereas determining an invocation sequence of update functions for each trigger element solves the problem.

The examiner also asserts that Mukherjee discloses a document generation system where update values associated with questions in a questionnaire form are used to obtain information to produce a customized document from a template at figures 3A-3L, 4 and column 2, line 19 through column 3, line 5. Specifically, the examiner asserts that a user is presented with a plurality of questions in a questionnaire form (figures 3A-3L), and a document (item 400b, figure 4) is generated therefrom. For clarity, appellant does not assert to have invented document generation systems (see the background of this application). Appellant was merely pointing out that the cited references are not particularly on point. Automated document generation systems typically generate a customized document such as a legal contract by using a template document and answers submitted in response to a questionnaire document. The template includes conditional and non-conditional content. The conditional content is selected by evaluating rules based on the values assigned to answers resulting from the questionnaire document. For example, a user without legal expertise would input answers to questions posed by the questionnaire, and those answers would be used to generate a customized contract document from the template by selecting particular clauses of legal significance (conditional content) to include or exclude along with other content that may always be included (non-conditional content). One of the advantages of such systems is reduced reliance on skilled personnel such as lawyers. In contrast, Mukherjee digitizes paper forms and attempts to eliminate irrelevant and redundant questions in a set of related forms. There is no conditional content in the forms so the forms are not template documents, and the result does not contain selected conditional content but simply the answers provided by the user so the result is not a customized document but rather just a completed form.

Conclusion

The rejections are improper for at least the reasons set forth above. Appellants accordingly request that the rejections be reversed and the application put forward for allowance.

Respectfully submitted,

/Holmes W. Anderson/
Holmes W. Anderson
Reg. No. 37,272
Attorney for Assignee

Date: March 7, 2011

Anderson Gorecki & Manaras LLP
33 Nagog Park
Acton MA 01720
(978) 264-4001

Appendix A - Claims

1. (previously presented) A method of generating an invocation sequence of update functions that update values associated with questions in a questionnaire form used to obtain information to produce a customized document from a template document, the invocation sequence specifying an order in which the update functions are invoked, the method comprising the steps of:

supplying the update functions to a synthesiser;

identifying, using the synthesizer, trigger questions from the questions of the questionnaire form that trigger invocation of at least one of the supplied update functions;

generating an activation network based on the identified update functions at the synthesiser; and

determining the invocation sequence of the identified update functions for each trigger question by using the activation network.

2. (original) The method of claim 1, wherein the trigger elements are determined by at least one of the value or status of the elements of the form.

3. (original) The method of claim 1, wherein the activation network includes cyclic update functions.

4. (original) The method of claim 1, 2, or 3 further comprising the step of:

exporting the update functions and the invocation sequence to a form renderer in a readable format.

5. (original) The method of claim 1, 2, or 3 wherein the update functions are validation functions.

6. (original) The method of claim 1, 2, or 3 wherein the update functions are activation functions.

7. (original) The method of claim 4, wherein the synthesiser is stored on a server computer.

8. (original) The method of claim 4, wherein the synthesiser is stored on a client computer.

9. (original) The method of claim 4, wherein the synthesiser forms part of a middleware application, located between a server computer and a client computer.

10. (original) The method of claim 4, wherein the synthesiser is integrated with the form renderer.

11. (original) The method of claim 10, wherein the form renderer is a web browser application.

12. (original) The method of claim 1, 2 or 3 wherein the update functions are supplied by one of a database engine and a form generator.

13. (original) The method of claim 1, 2 or 3 wherein the step of determining the invocation sequence involves determining the order in which the update functions must be executed within the activation network.

14. (previously presented) The method of claim 1, 2 or 3 further comprising the steps of:

entering data to change the status of a first form question;

determining the position of the first form question in the activation network; and

triggering the update functions associated with the first form question to update the status of a second form question.

15. (previously presented) A computer readable medium having program code stored thereon, which, when run on a computer, causes the computer to generate invocation sequences of update functions to update values associated with questions in a questionnaire form used to obtain information to produce a customized document from a template document by performing the steps of:

supplying the update functions to a synthesiser;

identifying, with the synthesizer, trigger questions from the questions of the form that trigger the invocation of ones of the supplied update functions;

generating an activation network based on the update functions identified at the synthesiser; and

determining the invocation sequence of update functions for each trigger question by using the activation network.

16. (previously presented) A computer readable medium having program code stored thereon for generating an invocation sequence to update values associated with questions in a questionnaire form, comprising:

a synthesiser for generating an activation network based on update functions supplied by a form generator and identifying trigger questions from the questions of the form that trigger the invocation of ones of the update functions; and

a determinator for determining the invocation sequence of the supplied update functions, for each trigger question, the form being updated based on the determined invocation sequence and output in tangible format.

17. (previously presented) The computer readable medium of claim 16, wherein the trigger questions are determined by at least one of the value or status of the questions of the form.

18. (previously presented) The computer readable medium of claim 16, wherein the activation network includes cyclic update functions.

19. (previously presented) The computer readable medium of claim 16, 17 or 18, wherein the update functions and invocation sequence are exported to a form renderer in a readable format.

20. (previously presented) The computer readable medium of claim 16, 17 or 18, wherein the update functions are validation functions.

21. (previously presented) The computer readable medium of claim 16, 17 or 18, wherein the update functions are activation functions.

22. (previously presented) The computer readable medium of claim 16, 17 or 18, wherein the synthesiser is stored on a server computer.

23. (previously presented) The computer readable medium of claim 16, 17 or 18, wherein the synthesiser is stored on a client computer.

24. (previously presented) The computer readable medium of claim 16, 17 or 18, wherein the synthesiser forms part of a middleware application, located between a server computer and a client computer.

25. (previously presented) The computer readable medium of claim 16, 17 or 18, wherein the synthesiser is integrated with the form renderer.

26. (previously presented) The computer readable medium of claim 25, wherein the form renderer is a web browser application.

27. (previously presented) The computer readable medium of claim 16, 17 or 18, wherein the update functions are supplied by one of a database engine and a form generator.

28. (previously presented) The computer readable medium of claim 16, 17 or 18, wherein the step of determining the invocation sequence involves determining the order in which the functions must be executed within the activation network.

29. (previously presented) The computer readable medium of claim 16, 17 or 18, wherein, in use, a user enters data to change the status of a first form question, the position in the activation network of the first form question is determined such that the update functions associated with the first form question are triggered so as to update the status of a second form question, in accordance with the invocation sequence.

Appendix B - Evidence Submitted

None.

Appendix C - Related Proceedings

None.